

Hierarchical community structure in complex (social) networks.*

EMANUELE MASSARO^{§,†}, FRANCO BAGNOLI[¶]

[§]Dept. of Information Engineering and CSDC, University of Florence, Italy

[†]Now at Dept. of Civil and Environmental Engineering
Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA

[¶]Dept. of Physics and Astronomy, and CSDC, University of Florence,
also INFN sez. Firenze, Italy

The investigation of community structure in networks is a task of great importance in many disciplines, namely physics, sociology, biology and computer science where systems are often represented as graphs. One of the challenges is to find local communities from a local viewpoint in a graph without global information in order to reproduce the subjective hierarchical vision for each vertex. In this paper we present the improvement of an information dynamics algorithm in which the label propagation of nodes is based on the Markovian flow of information in the network under cognitive-inspired constraints [1]. In this framework we have introduced two more complex heuristics that allow the algorithm to detect the multi-resolution hierarchical community structure of networks from a source vertex or communities adopting fixed values of model's parameters. Experimental results show that the proposed methods are efficient and well-behaved in both real-world and synthetic networks.

PACS numbers: PACS 07.05.Kf, 89.65.-s, 89.75.Fb

1. Introduction

We live in a world of networks. The networks around us and ourselves, as people, we are part of the network of social relations between individuals. Examples of networks in the world are the WWW, the rail network, the subway, neural networks, the telephone network, or less concrete entities, such as the relations of knowledge and collaboration between people. In general, a graph or network is a very general approximation of a system constituted

* Presented at SummerSoltice 2013

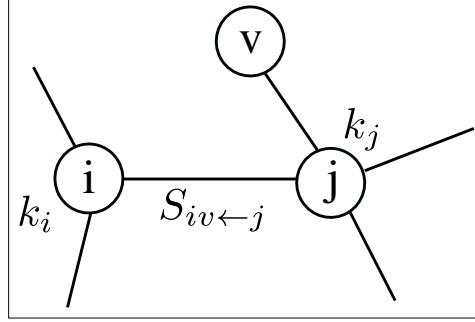


Figure 1. Communication phase, Let us consider the node i that communicates with the node j . The node j passes his knowledge of the world to the node i . Here we can consider three different scenarios: the first one takes into account that the knowledge depends on the connection degree of the incoming node. In this way the most connected nodes have less time to talk with others. In the second hypothesis the spread information depends of the connectivity degree of the selected nodes: then Eq. (1) becomes $S_{iv \leftarrow j} = mS_{iv} + (1 - m)S_{jk}/k_i$. In the third one it depends both on the degree of node i and on the degree of node j : $S_{iv \leftarrow j} = mS_{iv} + (1 - m)S_{jk}/\sqrt{k_i^2 + k_j^2}$.

by many entities, called nodes (persons, computers, proteins, chemicals, etc.) linked to each other and interacting through connections (which may be, therefore, a cable between computers, hyperlinks between web pages, a collaboration between people, a reaction between chemical substances, etc.). The study of networks is therefore very relevant in many disciplines, given the wide variety of structures and systems of the real world that can be incorporated into the category of “complex networks”. Therefore a complex network is a system with non trivial topological features and would normally not detectable in simple graphs. The majority of real networks, be they social, biological or technological may be considered complex: this depends on some features such as, eg., the nodes’ degree of distribution [2, 3], the assortativity between the vertices [4], a short path length [5, 6] and often an evident hierarchical structure [7]. Another important observed feature of complex networks is the presence of a community structure [8]. Many real networks are not homogeneous and not consist of a single block of indistinct nodes, rather they exhibit community structures, ie vertex groups where there are high densities of arches within each group, and a relatively low number of connections between the various groups. In recent years there has been explored in detail the possibility of automatically identify communities within networks, trough computation models, giving rise to a

new field of research called “community detection” [9, 10]. The problem of community detection has attracted the attention of various researchers coming from different disciplines from physics to social science. Nowadays no common definition for community has been agreed upon: it is widely accepted that a community is a group of vertices more linked than between the group and the rest of the graph. This is clearly a poor definition, and indeed, on a connected graph, there is not a clear distinction between a community and the rest of the graph. In general, there is a continuum of nested communities whose boundaries are somewhat arbitrary. Moreover in complex networks, and in particular in social networks, it is very difficult to give a clear definition of community: it is caused by the fact that nodes often results in overlapping communities because they belong to more than one cluster or module or community. For instance people usually belong to different communities at the same time, depending on their families, friends, colleagues, etc. [11]: so people, making a subjective screening of the network in which they live, have their local vision of communities to which they belong. Each of us knows perfectly what are their own communities: in this way we compute a local community detection algorithm revealing the multi-scale structure of the networks from a personal viewpoint. In order to have a subjective vision of the surrounding world it is necessary to develop community detection algorithms based on the local link structure of the network, that work by getting information from neighbours [12–17].

In this paper we show a method for detecting communities in networks where each node discovers its representation of the network by “talking” with the others and then elaborating the information through a cognitive-inspired mechanism [1]. In particular we introduce more accurate heuristics in order to allow the method to be free by the parameters m and α . The main feature of these two heuristics called respectively *IDA + LTE* and *Double pruning* is to reveal the hierarchical community structure of complex networks from a local viewpoint. We demonstrate that a node can discover the network’s multi-levels giving its subjective vision of the world until it wants or can explore the graph without any global information about it.

The reminder of the paper is organized as follows. First, in Section 2, the general method [1] is presented. Then we introduce two more complex heuristics applied to this model in order to reveal the multi-resolution community structure of both synthetic and real networks. In the results we also compare our method with other well-known community detection algorithm by evaluating the so-called Normalized Mutual Information (NMI) on LFR benchmarks [18]. The paper is closed by final remarks and conclusions.

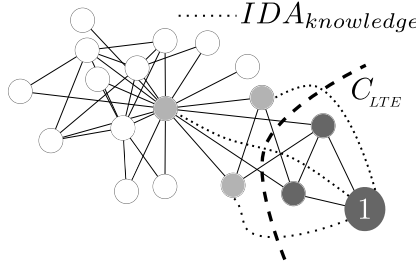


Figure 2. Schematic representation of IDA + LTE algorithm. We suppose that *LTE* detect for the node 1 the dark-gray nodes as community nodes (C_{LTE}): after that we compute the *IDA* on the selected nodes. The node takes the information from those accepted as its community creating a virtual link with these (dotted line in figure). The *LTE* decides whether to accept the new nodes members of own community or not.

2. The Method

We consider N individuals, labelled from 1 to N . Let us denote by A the adjacency matrix, $A_{ij} = 1$ (0) indicates the presence (absence) of a link from site j to site i . Each individual i is characterized by a state vector S_i , representing his knowledge of the outer world. We interpret S as a probability distribution, assuming that $S_i^{(v)}$ is the probability that individual i belongs to the community v . Thus, $S_i^{(v)}$ is normalized on the index v . We shall denote with $S = S(t)$ the state of the all network at time t , with $S_{iv} = S_i^{(v)}$. We shall initialize the system by setting $S_{ij}(0) = \delta_{ij}$, where δ is the Kroneker delta, $\delta_{ij} = 1$ if $i = j$ and zero otherwise.

In other words, at time 0 each node knows only about itself. In this implementation we assume that each individual spends the same amount of time in communicating, so the information depends on connection degree. Let us consider the node i that communicates with the node j as shown in Figure 1. We can assume that people with more connections dedicate less time to each of them. Since the amount of available time is limited we say that the knowledge of node i about node j depends on the connectivity degree of j , then $S_{i \leftarrow j}$ is function of the connectivity degree k of node j . Alternatively, we can consider that the information depends on the connectivity of i , in this way we give much more importance to the *speaking phase*.

The dynamics of the information is given by an alternation of communication and elaboration phases. Communication is implemented as a simple diffusion process, with memory m . The memory parameter m allows us to introduce some limitations in human cognitive such as the mechanism of oblivion and the timing effects: in fact the most recent informations have

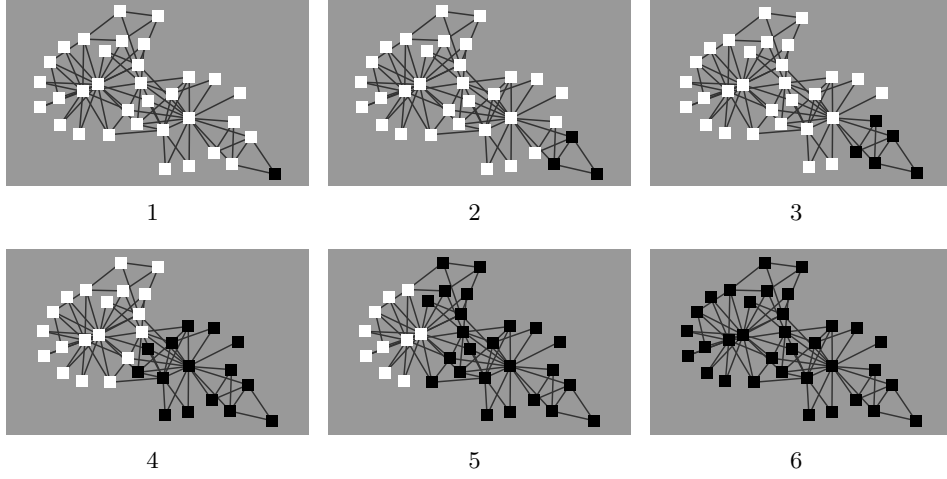


Figure 3. Node 17 in the *Zachary's Karate Club* network. In the first step the LTE algorithm accept 3 nodes for the local community. After that we compute the IDA + LTE algorithm: the knowledge of the other nodes permits to discover the structure of the whole network. From the starting vertex (black node at $T = 1$) at time $T = 2$ the community nodes are 17-6-7 (black nodes) and finally all the network is discovered by the node 17 thanks the information dynamics algorithm.

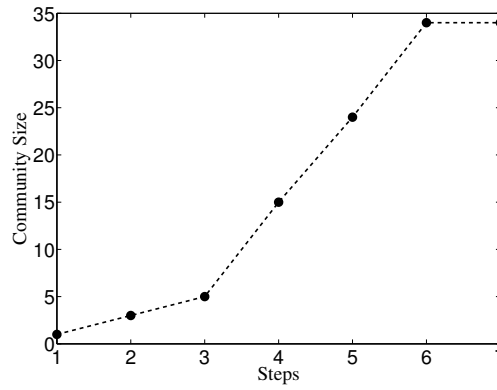


Figure 4. Temporal evolution (x -axis) of community size (y - axis) for the node 17 in the *Zachary's Karate Club* network.

more relevance than informations gained in the past. The knowledge of a generic node v for the node i trough j occurs via the communication phase

where the state of the system evolves as

$$S_{iv \leftarrow j} = mS_{iv} + (1 - m)\frac{S_{jv}}{k_j}. \quad (1)$$

The representation of the model is consistent with the ecological rationality of the system since the knowledge is distributed among the network elements in the form of bits of information (state vectors S_i). Our model is an improvement of the MCL algorithm proposed by Van Dongen [19] that simulates a random walk within the graph by an alternation of two phases called expansion and inflation. In our method we consider the Markov matrix $M_{i,j} = \frac{A_{ij}}{k_i}$, where k_i is the connectivity degree of node i . Finally, the evolution of the system is given by the sequence $S^0 \rightarrow S^1 \rightarrow S^2$. The communication phase based on the connectivity of incoming nodes, is given by the following:

$$S^1 = mS^0 + (1 - m)M'S^0, \quad (2)$$

where $0 < m < 1$ is a memory parameter and M' is the linear algebraic transpose of M . In this way the resulting matrix is stochastic again, i.e. the matrix elements (on each rows) correspond to probability values. The second phase, called elaboration or inflation is implemented as the rising of each element of the the matrix S_1 to the power α , where $\alpha > 1$ corresponds to the inflation parameter:

$$S_{ij}^2 = (S_{ij}^1)^\alpha \quad (3)$$

After this phase, in order to have a probability matrix we normalize the matrix S_2 over the columns. We assume that individuals have a large enough memory so that they can keep track of all information about all other individuals. In a real case, one should limit this memory and apply an input/output filtering.

3. Improvement of the method

In this section we want to explore the performance of our method taking into account two different heuristics in order to allow the method to be free by the parameters m and α . The main feature of these two heuristics called respectively *IDA + LTE* and *Double pruning* is to reveal the hierarchical community structure of complex networks from a local viewpoint. We demonstrate that a node can discover the network's multi-levels giving its subjective vision of the world until it wants or can explore the graph without any global information about the entire network. The experimental results show that the proposed algorithms are efficient in both real-world and synthetic networks.

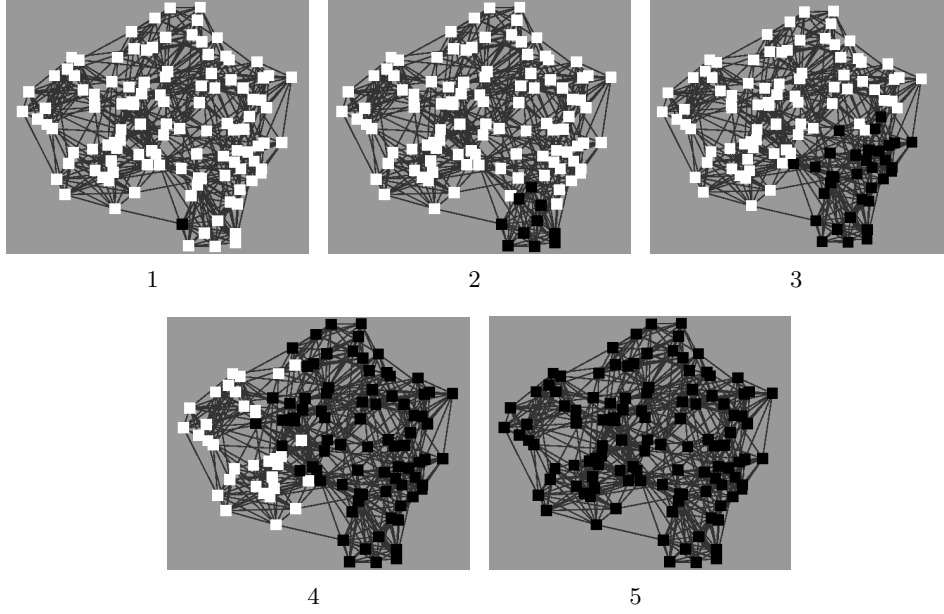


Figure 5. Washington (black node in 1) in NCAA football college network. Black nodes indicate the community of *Washington* for different time steps.

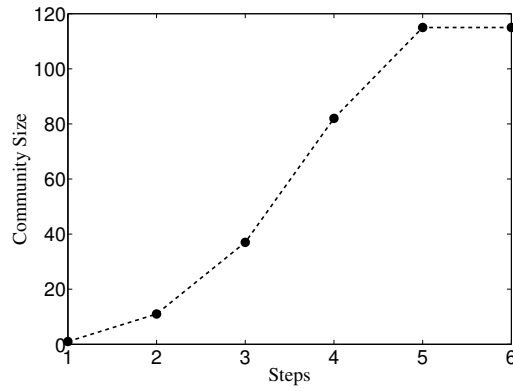


Figure 6. Temporal evolution (x -axis) of community size (y -axis) for Washington in the *NCCA College Football* network.

3.1. IDA + LTE

Recently Huang et al. [17] proposed a local algorithm for detecting communities in networks. Here we report a general description of this method,

known as Local Tightness Expansion Expansion (LTE), while it is well described in the Appendix. This method can reveal the community from a starting vertex via local optimization of the tightness measure which is proposed in ref [20], but the expansion is given by a tunable parameter which is not ecological in social networks where people elaborate the information through mental scheme or cognitive heuristics. The LTE algorithm can be summarized in the following:

1. Pick a vertex $s \in V$ as the starting vertex.
Let $C = \{S\}$ and $N = \Gamma(S) - \{S\}$
2. Select the vertex $a \in N$ that possess the largest similarity with vertices in C .
3. If $\tau_C^\beta(A) > 0$,
set $C = C \cup \{a\}$ and $N = N \cup \Gamma(a) - C$
4. Repeat 2 and 3 until $N = \emptyset$.

This algorithm, called local tightness expansion algorithm (LTE), is a very efficient local algorithm for detecting communities in networks from an individual point of view. Anyway when $N = \emptyset$ we arrive at a static convergence. Here we want to introduce a new method merging together our algorithm that we call Information Dynamics Algorithm (IDA) and LTE for analysing how the information dynamics can improve the performance of LTE and vice versa. We describe the new algorithm in the following (its representation is illustrated in Fig. 2):

1. Pick a vertex $s \in V$.
2. Run LTE and discover C_{LTE} for the vertex s .
3. Run IDA and take information of other nodes by community nodes
4. Run LTE and decide to accept new nodes.
5. Repeat 3 and 4 until a fixed memory or when the vertex s knows the whole network.

For testing purposes we use four real networks analysing and discussing our model peculiarities. The two case studies, of growing or different complexity, are the Zachary *karate club* network [21] as reported in the Fig. 3 and Fig. 4 and the NCAA college football network [22] as shown in the Fig. 5 and Fig. 6. For all the simulations we used always the same parameters. For the IDA we assumed $\alpha = 1.4$ and $m = 0.2$. For the LTE we used $\beta = 0.3$ for the first step and $\beta = 2$ during LTE+IDA dynamics. In this way we are able to discover the multi-resolution community structure of each node detecting the different local inner circles [23, 24].

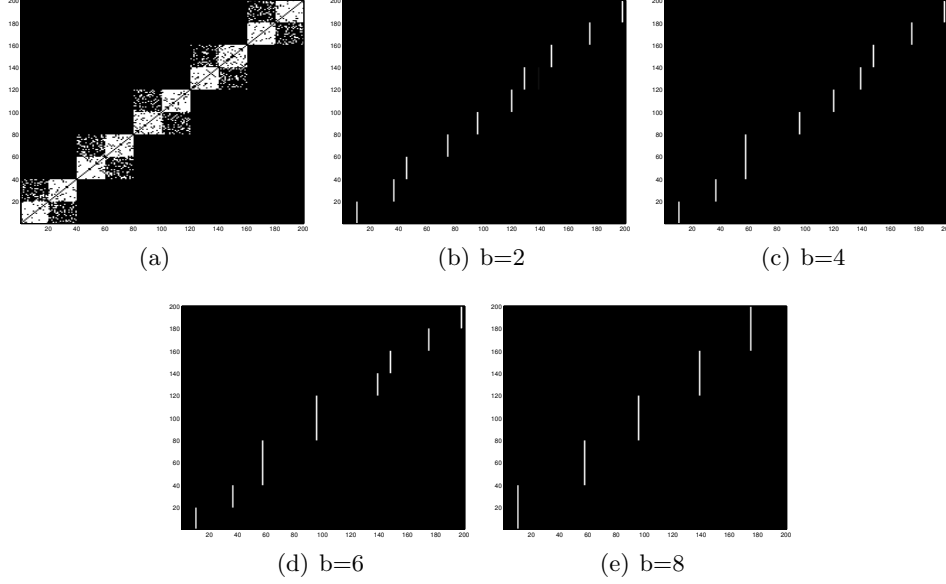


Figure 7. (a) Adjacency matrix of a network composed by 200 nodes and 3 levels where $p_1 = 0.9$, $p_2 = 0.2$ and $p_3 = 0$. Trough the double pruning heuristic it is possible to detect the different community levels reaching the discover of the four principal communities for $b = 8$.

3.2. Double pruning

Our model has two free parameters m and α and in previous works we have shown that it is very difficult to find specific values of these parameters in order to have a good representation of community distribution for different networks [1, 25, 26]. However, there is a simple procedure that can be used: at some moments for each node we evaluate the histogram of the state vector with 4 bins and we set to zero the values that are lower then the second bin values, as reported in the Algorithm 1. In this way we are able to generate different view of the clustering levels on hierarchical community structure networks as shown in Fig. 7. We use the Lancichinetti-Fortunato-Radicchi (LFR) benchmark graph [18, 27] to evaluate the accuracy of this method. We adopt the normalized mutual information (NMI) to evaluate the quality of detected communities which is currently widely used in measuring the performance of graph clustering algorithms [27]. The accuracy of our method is compared with three others well-known community detection algorithms. The observed results on our hierarchical networks allow us to state that for $b = 4$ we are able to detect the principal communities. For this reason for comparing our method with others we set $b = 4$.

Algorithm 1 pseudocode for double pruning heuristic

```

1:  $T, T_{max}, t_b = 0, t_1 = 0, c = 0, a = 0$ 
2:  $m = 0.2, \alpha = 1.4$ 
3: while  $t < T$  do
4:    $t = t + 1; b = b + 1;$ 
5:   if  $b < 2$  then
6:      $t_{max} = b * 50$ 
7:   else
8:      $t_{max} = b * 10$ 
9:   end if
10:  for  $t_1 = 1 : t_{max}$  do
11:     $a = a + 1;$ 
12:    if  $a = b$  then
13:      we evaluate the state vector of each node, then  $S^1$  is vector state of node 1,  $S^2$  of
14:      node 2 and so on...
15:      for  $i = 1 : N$  do
16:         $dx^i = (S_{max}^i - S_{min}^i)/3$ 
17:        for  $j = 1 : N$  do
18:          if  $S_{i,j} < S_{min}^i + 2 * dx^i$  then
19:             $S_{i,j} = 0$ 
20:          end if
21:        end for
22:      end for
23:      we normalize  $S$ 
24:    else
25:       $S_1 = mS_0 + (1 - m)M'S_0, S_2 = S_1^\alpha$ 
26:      we normalize  $S$ 
27:       $a = 0$ 
28:    end if
29:  end for
30:   $S = I(N)$ 
31: end while

```

In order to evaluate and partially validate our approach, we have applied our algorithm comparing its performance with 3 community detection algorithms namely *Infomap* [28], *Infomod* [29] and *MCL* [19] as reported in Fig.8. The input parameters of the benchmark graphs used here are: number of nodes $N = 1000$, average degree is $\hat{k} = 20$, maximum degree $\hat{k} = 50$ for all the networks. Moreover we changed the range of community size generating two kinds of networks of 1000 nodes: $1000S$ (S stay for small) means that communities have between 10 and 50 nodes and $1000B$ (B stay for big) means that communities have between 20 and 100 nodes. Results of the performance's comparison between our algorithm and the others are reported in Fig. 8. Our algorithm with the double pruning heuristic is very competitive with the other algorithms except for the Infomap method which is nowadays the best algorithm for detecting communities in static network, even if it can not be easily applied in dynamic environments.

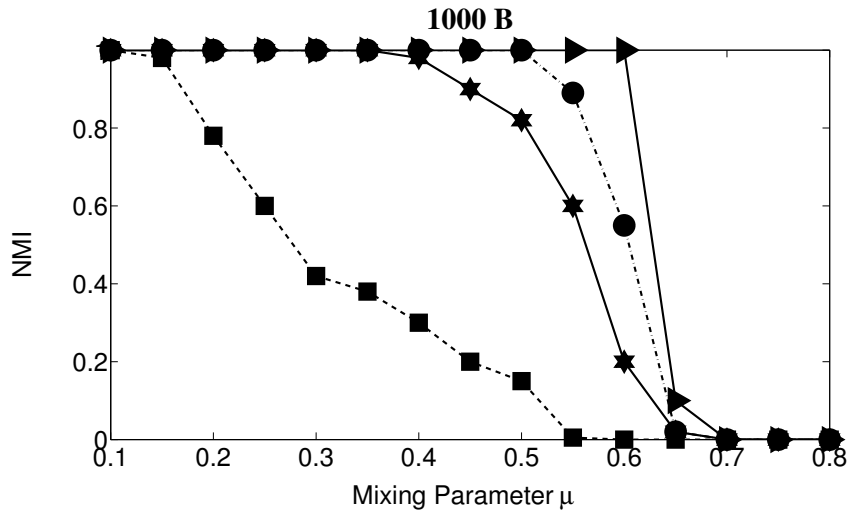
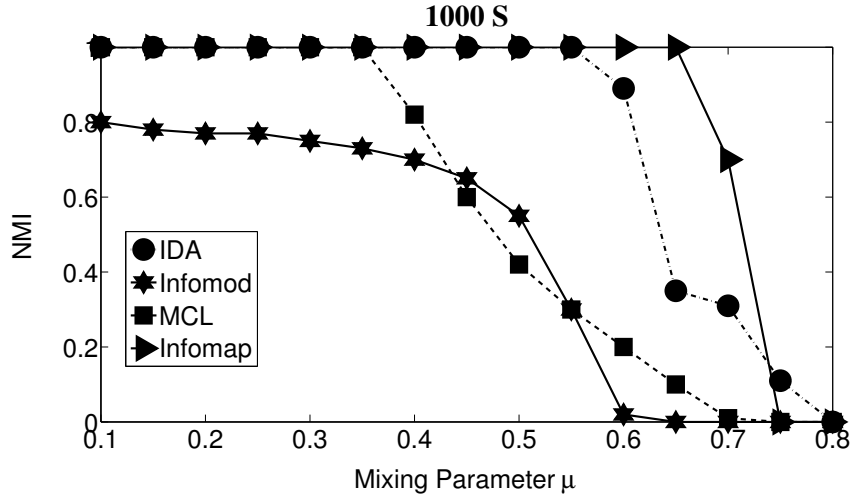


Figure 8. (a)-(b) Comparison between our method (IDA) and other algorithms (Infomod, MCL and Infomap) on the LFR Benchmark graphs with different size of communities respectively Small (a) and Big (b) increasing the mixing parameter μ .

4. Conclusions

In this paper we presented the state of the art of a community detection algorithm which is based on pure local information propagation inspired by the so-called Markov Clustering Algorithm (MCL) by S. Van Dongen [19]. The proposed method is able to identify communities both overlapping and non-overlapping, and we show that it is capable to identify communities also in dynamic networks [1, 25, 26]. Moreover, we describe that our algorithm is a multi-levels solution that can be used to capture the hierarchical community structure from a local viewpoint at any resolution. Experimental results on the real-world and synthetic datasets show that our algorithm achieves good performance. Concluding we show that the adaptation of two heuristics, namely *IDA+LTE* and *Double Pruning* are very effective to allow the method to be independent of the two parameters m and α .

References

- [1] E. Massaro, F. Bagnoli, A. Guazzini, and P. Lió, “Information dynamics algorithm for detecting communities in networks,” *Comm. Nonlin. Sci Numer. Simul.*, vol. 17, no. 11, pp. 4294 – 4303, 2012.
- [2] P. Erdős and A. Rényi, “On the evolution of random graphs,” in *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pp. 17–61, 1960.
- [3] A. L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [4] M. E. J. Newman, “Assortative mixing in networks,” *Phys. Rev. Lett.*, vol. 89, no. 20, p. 208701, 2002.
- [5] R. Albert, H. Jeong, and A. L. Barabasi, “The diameter of the world wide web,” *Nature*, vol. 401, pp. 130–131, 1999.
- [6] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, pp. 440–442, June 1998.
- [7] E. Ravasz and A. L. Barabási, “Hierarchical organization in complex networks,” *Phys. Rev. E*, vol. 67, no. 2, p. 026112, 2003.
- [8] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *PNAS*, vol. USA 99, p. 7821–7826, 2002.
- [9] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3–5, pp. 75 – 174, 2010.

- [10] M. E. J. Newman, “Detecting community structure in networks,” *The European Physical Journal B*, vol. 38, no. 2, pp. 321–333, 2004.
- [11] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, pp. 814–818, June.
- [12] J. P. Bragrow, “Evaluating local community methods in networks,” *Journal of Statistical Mechanics: Theory and Application*, p. P05001, 2008.
- [13] J. P. Bagrow and E. M. Bollt, “Local method for detecting communities,” *Phys. Rev. E*, vol. 72, p. 046108, Oct 2005.
- [14] A. Clauset, “Finding local community structure in networks,” *Phys. Rev. E*, vol. 72, p. 026132, Aug 2005.
- [15] F. Luo, J. Wang, and E. Promislow, “Exploring local community structures in large networks,” in *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pp. 233 –239, dec. 2006.
- [16] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.
- [17] J. Huang, H. Sun, Y. Liu, Q. Song, and T. Wenginger, “Towards online multiresolution community detection in large-scale networks,” *PLoS ONE*, vol. 6, no. 8, p. e23829, 2011.
- [18] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Phys. Rev. E*, vol. 78, p. 046110, Oct. 2008.
- [19] S. V. Dongen, *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.
- [20] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, “Defining and identifying communities in networks,” *PNAS*, vol. 101, p. 2658, 2004.
- [21] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *Journal of Anthropological Research*, no. 33, p. 452–473, 1977.

- [22] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson *Behavioral Ecology and Sociobiology*, no. 54, p. 396–405, 2003.
- [23] R. Dunbar, “Neocortex size as a constraint on group size in primates,”
- [24] R. Dunbar, “Coevolution of neocortical size, group size and language in humans,”
- [25] F. Bagnoli, E. Massaro, and A. Guazzini, “Community-detection cellular automata with local and long-range connectivity,” *LNCS, Springer, Berlin 2012*, vol. 7495, pp. 204–213, 2012.
- [26] D. Borkmann, A. Guazzini, E. Massaro, and S. Rudolph, “A cognitive-inspired model for self-organizing networks,” in *IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, pp. 229–234, 2012.
- [27] A. Lancichinetti and S. Fortunato, “Community detection algorithms: a comparative analysis,” 2009.
- [28] M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” *PNAS*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [29] M. Rosvall and C. Bergstrom, “An information-theoretic framework for resolving community structure in complex networks,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 18, p. 7327, 2007.

Appendix A

Appendix I

Usually, a network can be represented by a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. **Definition 1** (*Neighborhood*) Let $G = (V, E, w)$ be a weighted undirected network and $w(e)$ be the weight of the edge e . For a vertex $u \in V$, the structure neighbourhood of vertex u is the set $\Gamma(u)$ containing u and its adjacent vertices which are incident with a common edge with u : $\Gamma(u) = \{v \in V \mid \{u, v\} \in E\} \cup \{u\}$.

Definition 2 (*Structural Similarity*) Given a weighted undirected network $G = (V, E, w)$, the structure similarity $s(u, v)$ between two adjacent vertices u and v is:

$$s(u, v) = \frac{\sum_{x \in \Gamma(u) \cap \Gamma(v)} w(u, x) \cdot w(v, x)}{\sqrt{\sum_{x \in \Gamma(u)} w^2(u, x)} \cdot \sqrt{\sum_{x \in \Gamma(v)} w^2(v, x)}}. \quad (\text{A.1})$$

When we consider an unweighted graph, the weight $w(u, v)$ of any edge $\{u, v\} \in E$ can be set to 1 and the equation above can be transformed to:

$$s(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{|\Gamma(u)| \cdot |\Gamma(v)|}}. \quad (\text{A.2})$$

It corresponds to the so-called edge-clustering coefficient introduced by Radicchi et al. [20].

Definition 3 (*Tightness*) By employing the structural similarity, we introduce tightness, a new quality function of a local community C , which is given as follows:

$$T(C) = \frac{S_{in}^C}{S_{in}^C + S_{out}^C} \quad (\text{A.3})$$

where $S_{in}^C = \sum_{u \in C, v \in C, \{u, v\} \in E} s(u, v)$ is the internal similarity of the community C which is equal to two times of the sum of similarities between any two adjacent vertices both inside the community C ;

$S_{out}^C = \sum_{u \in C, v \in N, \{u, v\} \in E} s(u, v)$ is the external similarity of the community C which is equal to the sum of similarities between vertices inside the community C and vertices out of it.

The tightness measure is extended from the weak community definition proposed in Ref. [20]. Similar to other community definitions, the tightness

value of a community C , denoted by $T(C)$, will increase when sub-graph C has high internal similarity and low external similarity. The whole network without outward edges will achieve the maximal value 1, but the problem here is to find the local optimization of the measurement for each community. Suppose a community C is detected from a certain vertex s . We explore the adjacent vertices in the neighbourhood set N of S as shown in Fig. 2. So the variant tightness of the community $C \cup \{A\}$ becomes

$$\begin{aligned} T(C \cup \{A\}) &= \frac{S_{in}^C + 2S_{in}^a}{(S_{in}^C + S_{in}^a) + (S_{out}^c - S_{in}^a + S_{out}^a)} = \\ &= \frac{S_{in}^C + 2S_{in}^a}{(S_{in}^C + S_{in}^a + S_{out}^c + S_{out}^a)}, \end{aligned} \quad (A.4)$$

where $S_{in}^a = \sum_{\{v,a\} \in E \wedge v \in C} s(v, a)$; $S_{out}^a = \sum_{\{a,u\} \in E \wedge u \notin C} s(a, u)$. Then the tightness increment of a vertex a joining in C is:

$$\begin{aligned} \Delta T_C(A) &= T(C \cup \{A\}) = \\ &= \frac{S_{in}^C + 2S_{in}^a}{(S_{in}^C + S_{in}^a + S_{out}^c + S_{out}^a)} - \frac{S_{in}^C}{S_{in}^C + S_{out}^c} = \\ &= \frac{2S_{in}^a \cdot S_{out}^c - S_{in}^C \cdot S_{out}^a + S_{in}^C \cdot S_{in}^a}{(S_{in}^C + S_{in}^a + S_{out}^c + S_{out}^a)(S_{in}^C + S_{out}^c)} \end{aligned} \quad (A.5)$$

If $\Delta T_C(A) > 0$, then $2S_{in}^a \cdot S_{out}^c - S_{in}^C \cdot S_{out}^a + S_{in}^C \cdot S_{in}^a > 0$ which is equivalent to $\frac{S_{out}^c}{S_{out}^a} - \frac{S_{out}^a - S_{in}^a}{2S_{in}^a}$. Then they define the tightness gain in the following [17].

Definition 4 (Tightness Gain) The tightness gain for the community C adopting a neighbor vertex a can be denoted as

$$\tau_C(A) = \frac{S_{out}^c}{S_{in}^C} - \frac{S_{out}^a - S_{in}^a}{2S_{in}^a}. \quad (A.6)$$

It means that the ratio of external similarity to internal similarity of community C is greater than the ratio of external similarity increment to internal similarity increment caused by adopting vertex a . Obviously, this case will result in the increase of the tightness value of community C . Therefore, $\tau_C(a)$ can be utilized as a criterion to determine whether the candidate vertex a should be included in the community C or not. In the following, they introduce an optional resolution parameter β to control the scale at which we want to observe the communities in a network.

Definition 5 (*Tunable Tightness Gain*) *The tunable tightness gain for the community C merging a neighbor vertex a can be denoted as*

$$\tau_C^\alpha(A) = \frac{S_{out}^C}{S_{in}^C} - \frac{\alpha S_{out}^a - S_{in}^a}{2S_{in}^a}. \quad (\text{A.7})$$

A parameter $\beta \in (0, \infty)$ is introduced as the coefficient of S_{out}^a which can increase or decrease the proportion of the external similarity of the candidate vertex a . Here, the criterion for accepting a vertex a is changed to $\tau_C^\alpha(A) > 0$. For $\alpha = 1$, the criteria is moderate and can be used in most normal cases. In [17] the authors shows different scenarios for different values of the free parameter β : setting $\beta \in (0, 1)$, the value of S_{out}^a is reduced by this coefficient which increases the chance of a candidate vertex a joining C and bigger communities will be formed compared to the normal case with $\beta = 1$. On the contrary, it will result in the formation of smaller communities in a network when we set $\beta > 1$.